

Hash

$$n + e$$

Tsinghua University

2016 年 6 月 20 日



① 写法

存储

插入

查找

② 构建 Hash 函数

③ 解决冲突的方法

④ Task

① 写法

存储

插入

查找

② 构建 Hash 函数

③ 解决冲突的方法

④ Task

- 个人建议是用类似边表的结构, 方便插入和查找, 还能存储其他信息

```
struct E{//edge
    int key, next, num, ...;
}e[Maxn];
int last[Maxn], et = 0;
```

① 写法

存储

插入

查找

② 构建 Hash 函数

③ 解决冲突的方法

④ Task

- 将元素 x 经过函数处理后得到 key , 然后把 x 和 key 塞到边表里面

```
void insert (int x, int key) {  
    int i, flag = 0;  
    for (i = last[key]; i && !flag; i = e[i].next)  
        if (e[i].key == key) e[i].num++, flag = 1;  
    if (!flag) {  
        e[++et] = (E) {x, last[key], 1};  
        last[key] = et;  
    }  
}
```

① 写法

存储

插入

查找

② 构建 Hash 函数

③ 解决冲突的方法

④ Task

- ```
int find (int x, int key) {
 for (int i = last[key]; i; i = e[i].next)
 if (e[i].key == x) return 1;
 return 0;
}
```



① 写法

② 构建 Hash 函数

③ 解决冲突的方法

④ Task

- 现在是怎么构建一个函数，把  $x$  弄成  $key$  的问题了
  - (方法：直接寻址法、数字分析法、平方取中法、折叠法、随机数法、除留余数法) 中间四个就算了 (把这行无视掉) (它在瞎 BB)

- 现在是怎么构建一个函数，把  $x$  弄成  $key$  的问题了
  - (方法：直接寻址法、数字分析法、平方取中法、折叠法、随机数法、除留余数法) 中间四个就算了 (把这行无视掉) (它在瞎 BB)
- 对于
  - 数：直接找个大数 mod 了。这里如果数是随机的话，没有必要找一个质数。如果怕出事，平时随便找个质数顶上去就好了。戳我

- 现在是怎么构建一个函数，把  $x$  弄成  $key$  的问题了
  - (方法：直接寻址法、数字分析法、平方取中法、折叠法、随机数法、除留余数法) 中间四个就算了 (把这行无视掉) (它在瞎 BB)
- 对于
  - 数：直接找个大数 mod 了。这里如果数是随机的话，没有必要找一个质数。如果怕出事，平时随便找个质数顶上去就好了。戳我
  - 坐标 & DP 状态 & 搜索状态： $i \cdot n + j$ 、 $(i \cdot n + j) \cdot m + k$ 。表达式越简单越好

- 现在是怎么构建一个函数，把  $x$  弄成  $key$  的问题了
  - (方法：直接寻址法、数字分析法、平方取中法、折叠法、随机数法、除留余数法) 中间四个就算了 (把这行无视掉) (它在瞎 BB)
- 对于
  - 数：直接找个大数 mod 了。这里如果数是随机的话，没有必要找一个质数。如果怕出事，平时随便找个质数顶上去就好了。戳我
  - 坐标 & DP 状态 & 搜索状态： $i \cdot n + j$ 、 $(i \cdot n + j) \cdot m + k$ 。表达式越简单越好
  - 字符串 & 数组：转成  $H$  进制 ( $H$  可以取 26, 131, 13131...) 或者类  $H$  进制 (就是每个位置再加一个权)，转成一个数字取中间的一段：hash 要支持区间减法。演示？类似“差分”与“前缀和”

- 现在是怎么构建一个函数，把  $x$  弄成  $key$  的问题了
  - (方法：直接寻址法、数字分析法、平方取中法、折叠法、随机数法、除留余数法) 中间四个就算了 (把这行无视掉) (它在瞎 BB)
- 对于
  - 数：直接找个大数 mod 了。这里如果数是随机的话，没有必要找一个质数。如果怕出事，平时随便找个质数顶上去就好了。戳我
  - 坐标 & DP 状态 & 搜索状态： $i \cdot n + j$ 、 $(i \cdot n + j) \cdot m + k$ 。表达式越简单越好
  - 字符串 & 数组：转成  $H$  进制 ( $H$  可以取 26, 131, 13131...) 或者类  $H$  进制 (就是每个位置再加一个权)，转成一个数字取中间的一段：hash 要支持区间减法。演示？类似“差分”与“前缀和”
  - 图 & 树：找重心，最小表示法，... 不展开你们以后会明白的，现在暂时用不到

① 写法

② 构建 Hash 函数

③ 解决冲突的方法

④ Task

- 1. mod 的数变大 (int 改 long long, 效率变慢, 空间消耗变大)
- 2. 双 hash, 三 hash (同上, 至今不会算碰撞概率, 玄学)
- \*3. 增加信息 最无脑最好用



## 之前的一道题

- 2566 矩阵 hash: 元素多, 信息多, 碰撞概率大

## 之前的一道题

- 2566 矩阵 hash: 元素多, 信息多, 碰撞概率大? 个鬼

## 之前的一道题

- 2566 矩阵 hash: 元素多, 信息多, 碰撞概率大? 个鬼
- 要支持  $O(1)$  求出某个矩阵的 hash 值, 我想了想, 好像很麻烦. 也是类似二维矩阵前缀和的做法

## 之前的一道题

- 2566 矩阵 hash: 元素多, 信息多, 碰撞概率大? 个鬼
- 要支持  $O(1)$  求出某个矩阵的 hash 值, 我想了想, 好像很麻烦. 也是类似二维矩阵前缀和的做法
- 要支持  $O(n^2)$  求出所有边长为  $L$  的矩阵的 hash 值:  $O(1)$  转移即可
- 怎么好写怎么写: 我直接把每一行的数 xor ... 然后没事说好的表达式越简单越好连 “\*” 都没用到 2333

① 写法

② 构建 Hash 函数

③ 解决冲突的方法

④ Task

- 训练 2 (1638, 1390, 2087, 1495) 多与搜索结合
- 训练 4 (2138, 1964)
- 有些题目看起来解法很相似, 只不过是我想让你们巩固一下这类的想法, 并且与代码能力相关
- 应某些人要求多加了几道双向搜索的题目